



---

*Research article*

# A smoothing Levenberg-Marquardt algorithm for linear weighted complementarity problem

Panjie Tian, Zhensheng Yu\* and Yue Yuan

College of Science, University of Shanghai for Science and Technology, Shanghai 200093, China

\* **Correspondence:** Email : zhsh-yu@163.com.

**Abstract:** In this paper, we consider the solution of linear weighted complementarity problem (denoted by LWCP). Firstly, we introduce a new class of weighted complementary functions and show that its continuously differentiable. On this basis, the LWCP is reconstructed as a smooth system of equations, and then solved by the Levenberg-Marquardt method. The convergence of the algorithm is proved theoretically and numerical experiments are carried out. The comparative experiments show that the algorithm has some advantages in computing time and iteration times.

**Keywords:** linear weighted complementarity problem; smooth function; L-M method

**Mathematics Subject Classification:** 65K05, 90C33

---

## 1. Introduction

The Weighted Complementarity Problem (WCP), which is to find a pair of  $(x, s, y) \in R^n \times R^n \times R^m$  such that

$$x \geq 0, s \geq 0, xs = w, F(x, s, y) = 0, \tag{1.1}$$

where,  $F : R^{2n+m} \rightarrow R^{n+m}$  is a continuously differentiable function,  $w \in R_+^n$  is the given weight vector,  $xs := x \circ s$  is the componentwise product of the vectors  $x$  and  $s$ . When  $w = 0$ , WCP (1.1) reduces to the classical Nonlinear Complementarity Problem (NCP). At present, there are many effective algorithms [1–5] that can solve NCP. For examples, Newton method [1], Quasi-Newton

method [2], L-M method [3,4], Neural-Networks method [5] etc. If

$$F(x, s, y) = Px + Qs + Ry - a, \quad (1.2)$$

problem (1.1) is the Linear Weighted Complementarity Problem (LWCP) studied in this paper, which is to find a pair of  $(x, s, y) \in R^n \times R^n \times R^m$  such that

$$x \geq 0, s \geq 0, xs = w, Px + Qs + Ry = a, \quad (1.3)$$

where,  $P \in R^{(n+m) \times n}$ ,  $Q \in R^{(n+m) \times n}$ ,  $R \in R^{(n+m) \times m}$ ,  $a \in R^{n+m}$ , are given matrices and vector. In addition, when

$$F(x, s, y) = \begin{pmatrix} \nabla f(x) - s - A^T y \\ Ax - b \end{pmatrix}, \quad (1.4)$$

problem (1.1) is the perturbed Karush-Kuhn-Tucker(KKT) condition for the following Nonlinear Programming(NLP)

$$\min f(x), s. t. Ax = b, x \geq 0. \quad (1.5)$$

Problem(1.3) was introduced by Potra [6] in 2012 and has been widely studied for its important applications in management, market equilibrium, etc. Many equilibrium problems can also be transformed into LWCP to solve, such as the famous Fisher market equilibrium problem [7], and the quadratic programming and weighted center problem [6].

In recent years, many effective algorithms have been proposed to solve problem (1.1) or (1.3) [8–13]. For examples, Chi et al. [9] proposed the full-Newton step infeasible interior-point method for solving LWCP. Zhang et al. [12] proposed the smoothing Newton type method for solving LWCP. Tang et al. [13] proposed the nonmonotone L-M method for NWCP. The interior point method depends on the choice of initial value. The classical Newton method needs the positive definite of Hessian matrix, otherwise, it is difficult to guarantee that the Newton direction is descending. The L-M method does not depend on the choice of initial values, nor does it require the positive definiteness of the Hessian matrix. Therefore, this paper mainly considers using L-M method to solve problem (1.3). Motivated by [13], we consider using a nonmonotone L-M method to solve LWCP.

LWCP is a more general complementary model. For the solution of this model, we hope to use the WCP functions obtained by the extension of NCP functions. However, due to the existence of weighting term, not all NCP functions can be directly extended to WCP functions. For NCP function in the form of FB function, many scholars have extended it to WCP function. In this paper, motivated by the smoothed penalty function for [14], we construct a smoothing function for WCP. And then use L-M method to approximate the equivalent reconstruction equations of problem (1.3). The comparison experiment of random generation shows the feasibility and effectiveness of our algorithm.

The following notations will be used throughout this paper. The superscript  $T$  denotes transpose.  $R$  denotes real numbers,  $R^n$  represents the set of all  $n$  dimensional real column vectors. The matrix  $I$  denotes the identity matrix, and  $\|\cdot\|$  denotes 2-norm. All vectors in this article are column vectors.

## 2. Weighted complementary function and its properties

In this section, we study a class of complementary functions with participation weights and discuss its properties. Based on this weighted complementary function, the equivalent reconstruction equations of problem (1.3) are given.

**Definition 2.1.** For a fixed  $c \geq 0$ , a function  $\phi: R^2 \rightarrow R$  is called a weighted complementarity function [13], if it satisfies

$$\phi^c(a, b) = 0 \Leftrightarrow a \geq 0, b \geq 0, ab = c. \quad (2.1)$$

When  $c = 0$ ,  $\phi^c(a, b)$  reduces to the NCP function.

In this paper, to solve the LWCP (1.3), we hope to use the WCP functions obtained by the extension of NCP functions. However, due to the existence of weighting term, not all NCP functions can be directly generalized to WCP functions. For example, the two piecewise NCP functions given in [2]:

$$\phi(a, b) = \begin{cases} 3a - \left(\frac{a^2}{b}\right), & b \geq a > 0, \text{ or } 3b > -a \geq 0; \\ 3a - \left(\frac{b^2}{a}\right), & a > b > 0, \text{ or } 3a > -b \geq 0; \\ 9a + 9b, & \text{else.} \end{cases} \quad (2.2)$$

$$\phi(a, b) = \begin{cases} k^2 a, & b \geq k|a|; \\ 2kb - \left(\frac{b^2}{a}\right), & a > \frac{|b|}{k}; \\ 2k^2 a + 2kb + \left(\frac{b^2}{a}\right), & a < -\frac{|b|}{k}; \\ k^2 a + 4kb, & b \leq -k|a|. \end{cases} \quad (2.3)$$

For FB function, many scholars have extended it to WCP function. For example, Liu et al. [11] based on the symmetric disturbance FB function in [15] constructed:

$$\phi_c(\mu, a, b) = (1 + \mu)(a + b) - \sqrt{(a + \mu b)^2 + (\mu a + b)^2 + 2c + 2\mu^2}, \quad (2.4)$$

where,  $c$  is a given nonnegative vector.

Zhang[12] proposed:

$$\phi_\theta(\mu, a, b, c) = \sqrt{a^2 + b^2 - 2\theta ab + 2(1 + \theta)c + 2\mu} - a - b, \quad (2.5)$$

where,  $\theta \in (-1, 1]$ ,  $c$  is a given nonnegative vector.

In addition, [13] provides another smooth function:

$$\phi_{\tau,q}^c(a,b) = (a+b)^q - \left( \sqrt{a^2 + b^2 + (\tau-2)ab + (4-\tau)c} \right)^q, \quad (2.6)$$

where,  $c$  is a given nonnegative vector,  $\tau \in [0,4)$  is a constant,  $q > 1$  is an odd integer. Compared with (2.4) and (2.5), (2.6) does not need to introduce the smoothing factor  $\mu$ . By controlling the value of  $q$ , smoothing can be achieved. This smoothing method will be used to smooth the new WCP function given below.

$$\phi_{\tau}^c(a,b) = a+b - \sqrt{\tau(a-b)^2 + (1-\tau)(a^2 + b^2) + 2(1+\tau)c}, \quad (2.7)$$

where,  $c$  is a given nonnegative vector,  $\tau \in [0,1]$  is a constant.

Since Eq (2.7) is not smooth, we make the following smoothing treatment:

$$\phi_{\tau,q}^c(a,b) = (a+b)^q - \left( \sqrt{\tau(a-b)^2 + (1-\tau)(a^2 + b^2) + 2(1+\tau)c} \right)^q, \quad (2.8)$$

where,  $c$  is a given nonnegative vector,  $\tau \in [0,1]$  is a constant,  $q > 1$  is an odd integer.

**Theorem 2.1.** Let  $\phi_{\tau,q}^c$  be defined by (2.8) with  $\tau \in [0,1]$  and  $q > 1$  being a positive odd interger. Then  $\phi_{\tau}^q$  is a family of WCP functions, i.e.,

$$\phi_{\tau,q}^c(a,b) = 0 \Leftrightarrow a \geq 0, b \geq 0, ab = c. \quad (2.9)$$

*Proof.* Since for any  $\alpha, \beta \in R$  and any positive odd interger  $q$ , there is  $\alpha^q = \beta^q \Leftrightarrow \alpha = \beta$ . So we have

$$\begin{aligned} \phi_{\tau,q}^c(a,b) = 0 &\Leftrightarrow (a+b)^q = \left( \sqrt{\tau(a-b)^2 + (1-\tau)(a^2 + b^2) + 2(1+\tau)c} \right)^q \\ &\Leftrightarrow a+b = \sqrt{\tau(a-b)^2 + (1-\tau)(a^2 + b^2) + 2(1+\tau)c} \\ &\Leftrightarrow \phi_{\tau}^c(a,b) = 0. \end{aligned} \quad (2.10)$$

That is to say, we only need to prove that  $\phi_{\tau}^c(a,b)$  is a family of WCP functions. On the one hand, we fist suppose that  $\forall a, b \in R$  satisfy,  $\phi_{\tau}^c(a,b) = 0$  i.e.,

$$\sqrt{\tau(a-b)^2 + (1-\tau)(a^2 + b^2) + 2(1+\tau)c} = a+b. \quad (2.11)$$

By squaring the two sides of (2.11), we have  $2(1+\tau)ab = 2(1+\tau)c$ , which together with  $\tau \in [0,1]$  yields  $ab = c$ . By substituing  $ab = c$  into (2.2), we have  $\sqrt{a^2 + b^2 + 2ab} = a+b \geq 0$ . Since  $c = ab \geq 0$ , it follows that  $a \geq 0, b \geq 0$ . On the other hand, we suppose that  $a \geq 0, b \geq 0, ab = c$ , then  $a+b \geq 0$  and

$$\sqrt{\tau(a-b)^2 + (1-\tau)(a^2 + b^2) + 2(1+\tau)c} = \sqrt{a^2 + b^2 + 2ab} = |a+b| = a+b. \quad (2.12)$$

Which implies that  $\phi_{\tau}^c(a,b) = 0$ .

**Lemma 2.1.** Let  $\phi_{\tau,q}^c$  be defined by (2.8) with  $\tau \in [0,1]$  and  $q > 1$  being a positive odd interger. Let

$$h_\tau^c(a, b) = \sqrt{\tau(a-b)^2 + (1-\tau)(a^2 + b^2) + 2(1+\tau)c}. \quad (2.13)$$

Then

(i) When  $q > 1$ ,  $\phi_{\tau,q}^c$  is continuously differentiable at any  $(a, b) \in R^2$  with

$$\nabla \phi_{\tau,q}^c = \begin{bmatrix} \frac{\partial \phi_{\tau,q}^c}{\partial a} \\ \frac{\partial \phi_{\tau,q}^c}{\partial b} \end{bmatrix}, \quad (2.14)$$

where

$$\begin{aligned} \frac{\partial \phi_{\tau,q}^c}{\partial a} &= q[(a+b)^{q-1} - h_\tau^c(a, b)^{q-2}(a-\tau b)], \\ \frac{\partial \phi_{\tau,q}^c}{\partial b} &= q[(a+b)^{q-1} - h_\tau^c(a, b)^{q-2}(b-\tau a)]. \end{aligned}$$

(ii) When  $q > 3$ ,  $\phi_{\tau,q}^c$  is twice continuously differentiable at any  $(a, b) \in R^2$  with

$$\nabla^2 \phi_{\tau,q}^c(a, b) = \begin{bmatrix} \frac{\partial^2 \phi_{\tau,q}^c}{\partial a^2} & \frac{\partial^2 \phi_{\tau,q}^c}{\partial a \partial b} \\ \frac{\partial^2 \phi_{\tau,q}^c}{\partial b \partial a} & \frac{\partial^2 \phi_{\tau,q}^c}{\partial b^2} \end{bmatrix}, \quad (2.15)$$

where

$$\begin{aligned} \frac{\partial^2 \phi_{\tau,q}^c}{\partial a^2} &= q \left\{ (q-1)(a+b)^{q-2} - h_\tau^c(a, b, c)^{q-4} [(q-2)(a-\tau b)^2 + h_\tau^c(a, b, c)^2] \right\}, \\ \frac{\partial^2 \phi_{\tau,q}^c}{\partial b^2} &= q \left\{ (q-1)(a+b)^{q-2} - h_\tau^c(a, b)^{q-4} [(q-2)(b-\tau a)^2 + h_\tau^c(a, b)^2] \right\}, \end{aligned}$$

$$\frac{\partial^2 \phi_{\tau,q}^c}{\partial a \partial b} = \frac{\partial^2 \phi_{\tau,q}^c}{\partial b \partial a} = q \left\{ (q-1)(a+b)^{q-2} - h_\tau^c(a, b)^{q-4} [(q-2)(a-\tau b)(b-\tau a) - \tau h_\tau^c(a, b)^2] \right\}.$$

**Lemma 2.2.** Let  $\phi_{\tau,q}^c$  be defined by (2.8) with  $\tau \in [0, 1]$  and  $q > 1$  being a positive odd interger. Defining the closed and convex set  $\Omega(u) := \{u \in R^2 \mid \|u\| \leq \theta\}$ , where  $\theta$  is a positive constant. Then:

(i) When  $q > 1$ ,  $\phi_{\tau,q}^c$  is Lipschitz continuous on  $\Omega(u)$  for any  $\theta > 0$ .

(ii) When  $q > 3$ ,  $\nabla \phi_{\tau,q}^c$  is Lipschitz continuous on  $\Omega(u)$  for any  $\theta > 0$ .

Since  $\phi_{\tau,q}^c$  and  $\nabla \phi_{\tau,q}^c$  are bounded on the set  $\Omega(u)$ , therefore the conclusion (i) and (ii) can be obtained from the Mean-Value-Theorem.

Given weight vector  $w \in R_+^n$ , let  $z := (x, s, y) \in R^{2n+m}$  and

$$H(z) = H(x, s, y) := \begin{pmatrix} F(x, s, y) \\ \Phi_{\tau, q}^w(x, s) \end{pmatrix}, \quad (2.16)$$

where

$$\Phi_{\tau, q}^w(x, s) = \begin{pmatrix} \phi_{\tau, q}^{w_1}(x_1, s_1) \\ \vdots \\ \phi_{\tau, q}^{w_n}(x_n, s_n) \end{pmatrix}. \quad (2.17)$$

Then the solution of LWCP (1.3) is equivalent to the approximate solution of the system of equations  $H(z) = 0$ .

**Lemma 2.3.** Let  $H(z) : R^{2n+m} \rightarrow R^{2n+m}$ ,  $\Phi_{\tau, q}^w : R^{2n} \rightarrow R^n$  be defined by (2.16) and (2.17), respectively.

Then:

- (i)  $\Phi_{\tau, q}^w(x, s)$  is continuously differentiable at any  $z = (x, s, y) \in R^{2n+m}$ .
- (ii)  $H(z)$  is continuously differentiable at any  $z = (x, s, y) \in R^{2n+m}$  with its Jacobian

$$H'(z) = \begin{pmatrix} F'_x & F'_s & F'_y \\ D_1 & D_2 & 0 \end{pmatrix}, \quad (2.18)$$

where

$$\begin{aligned} D_1 &= \text{diag} \left\{ q \left[ (x_i + s_i)^{q-1} - h_{\tau}^{w_i}(x_i, s_i)^{q-2} (x_i - \tau s_i) \right] \right\}, i = 1, 2, \dots, n. \\ D_2 &= \text{diag} \left\{ q \left[ (x_i + s_i)^{q-1} - h_{\tau}^{w_i}(x_i, s_i)^{q-2} (s_i - \tau x_i) \right] \right\}, i = 1, 2, \dots, n. \\ h_{\tau}^{w_i}(x_i, s_i) &= \sqrt{\tau(x_i - s_i)^2 + (1 - \tau)(x_i^2 + s_i^2) + 2(1 + \tau)w_i}, i = 1, 2, \dots, n. \end{aligned}$$

Let  $H(z)$  be defined by (2.16), then its value function  $M : R^{2n+m} \rightarrow R_+$  can be defined as:

$$M(z) := \frac{1}{2} \|H(z)\|^2. \quad (2.19)$$

Obviously, the solution of LWCP (1.3) is also equivalent to the approximate solution of the system of equations  $M(z) = 0$ . In addition, the following conclusion can be obtained from the Lemma 2.3.

**Lemma 2.4.** Let  $M : R^{2n+m} \rightarrow R_+$  be defined by (2.19), then  $M(z)$  is continuously differentiable at any  $z \in R^{2n+m}$ , and  $\nabla M(z) = H'(z)^T H(z)$ .

### 3. Algorithm and convergence analysis

In this section, based on the WCP function in Section 2, we will give the smooth L-M type algorithm and its convergence.

**Algorithm 3.1** (A smooth L-M method)

**Step 0:** Choose  $\theta, \sigma, \gamma, \delta \in (0, 1)$  and  $z^0 := (x^0, s^0, y^0) \in R^{2n+m}$ , let  $0 \leq \varepsilon \leq 1$ , and  $C_0 = M(z^0)$ . Choose a sequence  $\{\eta_k \mid \forall k \geq 0, \eta_k \in (0, 1)\}$ , set  $k := 0$ .

**Step 1:** Compute  $H(z^k)$ . If  $\|H(z^k)\| \leq \varepsilon$  then stop.

**Step 2:** Let  $\mu_k := \theta \|H(z^k)\|^2$ . Compute the search direction  $d_k \in R^{2n+m}$  by

$$\nabla M(z^k) + (H'(z^k)^T H'(z^k) + \mu_k I) d_k = 0. \quad (3.1)$$

**Step 3:** If  $d_k$  satisfies

$$\|H(z^k + d_k)\| \leq \sigma \|H(z^k)\|. \quad (3.2)$$

Then let  $\alpha_k := 1$ , and go to step 5. Otherwise, go to step 4.

**Step 4:** Set  $j_k$  be the smallest nonnegative integer  $j$  satisfying

$$M(z^k + \delta^j d_k) \leq C_k - \gamma \|\delta^j d_k\|^2. \quad (3.3)$$

let  $\alpha_k := \delta^{j_k}$ , and go to step 5.

**Step 5:** Set  $z^{k+1} := z^k + \alpha_k d_k$  and

$$Q_{k+1} := \eta_k Q_k, C_{k+1} := \frac{\eta_k Q_k C_k + M(z^{k+1})}{Q_{k+1}}. \quad (3.4)$$

**Step 6:** Let  $k := k + 1$ , and go to step 1.

Existing L-M type methods [16–18] are usually designed based on the Armijo line search. While algorithm 3.1 adopts a nonmonotone derivate free line search. The choice of  $\eta_k$  controls the degree of nonmonotonicity. If  $\eta_k \equiv 0$ , then the line search is monotone.

**Theorem 3.1.** Let  $\{z^k\}$  be the sequence generated by Algorithm 3.1. Then,  $\{z^k\}$  satisfying  $M(z^k) \leq C_k$  for all  $k \geq 0$ .

*Proof.* By Algorithm 3.1  $C_0 = M(z^0)$ . We first assume that  $M(z^k) \leq C_k$ . If  $\nabla M(z^k) = 0$ , then Algorithm 3.1 terminates. Otherwise  $\nabla M(z^k) \neq 0$  which implies that  $H(z^k) \neq 0$ , hence  $\mu_k = \theta \|H(z^k)\|^2 > 0$ . So the matrix  $H'(z^k)^T H'(z^k) + \mu_k I$  is positive definite. Thus the search direction  $d_k$  in step 3 is well-defined and  $d_k \neq 0$ . Since  $\nabla M(z^k) \neq 0$ , we have

$$\nabla M(z^k)^T d_k = -d_k^T (H'(z^k)^T H'(z^k) + \mu_k I) d_k < 0. \quad (3.5)$$

This implies that  $d_k$  is a descent direction of  $M(z^k)$  at the point  $z^k$ . Next we will prove that at least

one step size is obtained by step 4. Inversely, we assume that for any  $j$ ,  $M(z^k + \delta^j d_k) > C_k - \gamma \|\delta^j d_k\|^2$ , then

$$M(z^k + \delta^j d_k) > C_k - \gamma \|\delta^j d_k\|^2 \geq M(z^k) - \gamma \|\delta^j d_k\|^2, \quad (3.6)$$

thereby

$$\frac{M(z^k + \delta^j d_k) - M(z^k) + \gamma \|\delta^j d_k\|^2}{\delta^j} > 0. \quad (3.7)$$

By letting  $j \rightarrow \infty$  in (3.7), we have  $\nabla M(z^k)^T d_k \geq 0$ , which contradicts (3.5). Therefore, we can always get  $z^{k+1}$  by Step 3 or Step 4. If  $z^{k+1}$  is generated by step 3, i.e.,  $\|H(z^k + d_k)\| \leq \sigma \|H(z^k)\|$ , then  $\frac{1}{2} \|H(z^k + d_k)\|^2 \leq \frac{1}{2} \sigma^2 \|H(z^k)\|^2$ , so  $M(z^{k+1}) \leq \sigma^2 M(z^k)$ . And because,  $\sigma \in (0, 1)$ , therefore, we have  $M(z^{k+1}) \leq \sigma^2 M(z^k) < M(z^k) \leq C_k$ . If  $z^{k+1}$  is generated by step 4, we can get  $M(z^{k+1}) \leq C_k$  directly. So, from (3.4), we can get that  $C_k \geq \frac{\eta_k Q_k M(z^{k+1}) + M(z^{k+1})}{Q_{k+1}} = M(z^{k+1})$ .

Hence, we conclude that  $M(z^k) \leq C_k$  for all  $k \geq 0$ .

Next, we first suppose that  $\nabla M(z^k) \neq 0$  for all  $k \geq 0$ . In order to discuss the convergence of algorithm 3.1, we need the following lemma.

**Lemma 3.1.** Let  $\{z^k\}$  be the sequence generated by Algorithm 3.1, then there exists a nonnegative constant  $C^*$  such that

$$\lim_{k \rightarrow \infty} M(z^k) = \lim_{k \rightarrow \infty} C_k = C^*. \quad (3.8)$$

*Proof.* By Theorem 3.1, we can get  $0 \leq M(z^k) \leq C_k$  for all  $k \geq 0$  and  $C_{k+1} \leq \frac{\eta_k Q_k C_k + C_k}{Q_{k+1}} = C_k$ . Hence,

by The Monotone Bounded Theorem, there exists a nonnegative constant  $C^*$  such that  $\lim_{k \rightarrow \infty} C_k = C^*$ .

By the definition of  $Q_k$ , we have

$$Q_{k+1} = 1 + \sum_{i=0}^k \prod_{j=0}^i \eta_{k-j} \leq 1 + \sum_{i=0}^k \eta_{\max}^{i+1} \leq \sum_{i=0}^{\infty} \eta_{\max}^i = \frac{1}{1 - \eta_{\max}}. \quad (3.9)$$

Hence, we conclude that  $\eta_k Q_k \leq \frac{\eta_{\max}}{1 - \eta_{\max}}$  is bounded, which together with  $\lim_{k \rightarrow \infty} C_k = C^*$  yields

$\lim_{k \rightarrow \infty} \eta_{k-1} Q_{k-1} (C_k - C_{k-1}) = 0$ . So, it follows from (3.4) that



$$\begin{aligned} M(z^{k+1}) &= Q_{k+1}C_{k+1} - \eta_k Q_k C_k = (\eta_k Q_k + 1)C_{k+1} - \eta_k Q_k C_k \\ &= \eta_k Q_k (C_{k+1} - C_k) + C_{k+1}. \end{aligned} \quad (3.10)$$

Hence

$$\lim_{k \rightarrow \infty} M(z^k) = \lim_{k \rightarrow \infty} [\eta_{k-1} Q_{k-1} (C_k - C_{k-1}) + C_k] = C^*. \quad (3.11)$$

We complete the proof.

**Theorem 3.2.** Let  $\{z^k\}$  be the sequence generated by Algorithm 3.1. Then any accumulation point  $z^*$  of  $\{z^k\}$  is a stationary point of  $M(z)$ .

*Proof.* By Lemma 3.1, we have  $\lim_{k \rightarrow \infty} M(z^k) = \lim_{k \rightarrow \infty} C_k = C^*$ ,  $C^* \geq 0$ . If  $C^* = 0$ , then  $\lim_{k \rightarrow \infty} H(z^k) = 0$  which together with Lemma 2.4 yields  $\nabla M(z^*) = 0$ . In the following, we discuss the case of  $C^* > 0$ . Set  $N := \left\{k \mid \|H(z^k + d_k)\| \leq \sigma \|H(z^k)\|\right\}$ . Then  $N$  must be a finite set, otherwise  $M(z^{k+1}) \leq \sigma^2 M(z^k)$  holds for infinitely many  $k$ . By letting  $k \rightarrow \infty$  with  $k \in N$ , we can have  $C^* \leq \sigma^2 C^*$  and  $1 \leq \sigma^2$  which contradicts  $\sigma \in (0, 1)$ . Therefore, we can suppose that there exists an index  $\bar{k} > 0$  such that  $\|H(z^k + d_k)\| > \sigma \|H(z^k)\|$  for all  $k \geq \bar{k}$ . Thereby, there exists a  $j_k$  such that  $M(z^{k+1}) \leq C_k - \gamma \|\delta^{j_k} d_k\|^2$ , i.e.,

$$\gamma \|\delta^{j_k} d_k\|^2 \leq C_k - M(z^{k+1}). \quad (3.12)$$

Next, we suppose that  $z^*$  is the limit of the subsequence  $\{z^k\}_{k \in K} \subset \{z^k\}$  where  $K \in \{0, 1, 2, \dots\}$ , i.e.,

$\lim_{k(\in K) \rightarrow \infty} z^k = z^*$ . Hence, by the continuity, we have  $C^* = M(z^*) = \frac{1}{2} \|H(z^*)\|^2$ . By  $\lim_{k \rightarrow \infty} \mu_k = \lim_{k \rightarrow \infty} \theta \|H(z^k)\|^2 = \lim_{k \rightarrow \infty} 2\theta M(z^k) = 2\theta C^*$ , we can get that

$$\lim_{k(\in K) \rightarrow \infty} \left[ H'(z^k)^T H'(z^k) + \mu_k I \right] = H'(z^*)^T H'(z^*) + 2\theta C^* I. \quad (3.13)$$

According to the proof process of theorem 3.1, the matrix  $H'(z^k)^T H'(z^k) + \mu_k I$  is a symmetric positive definite matrix. In addition, because of  $C^* > 0$ , the matrix  $H'(z^*)^T H'(z^*) + 2\theta C^* I$  is also symmetric positive definite matrix. Hence, we have

$$\lim_{k(\in K) \rightarrow \infty} \left[ H'(z^k)^T H'(z^k) + \mu_k I \right]^{-1} = \left[ H'(z^*)^T H'(z^*) + 2\theta C^* I \right]^{-1}. \quad (3.14)$$

and

$$\lim_{k(\in K) \rightarrow \infty} d_k = d^* = -\left[ H'(z^*)^T H'(z^*) + 2\theta C^* I \right]^{-1} \nabla M(z^*). \quad (3.15)$$

By (3.5), we can get

$$\nabla M(z^*)^T d^* = \lim_{k(\in K) \rightarrow \infty} \nabla M(z^k)^T d^k \leq 0. \quad (3.16)$$

By letting  $k \rightarrow \infty$  with  $k \in N$  in (3.12), we have  $\lim_{k(\in K) \rightarrow \infty} \|\delta^{j_k} d_k\| = 0$ . If  $\delta^{j_k} > 0$ , then

$\lim_{k(\in K) \rightarrow \infty} d_k = d^* = 0$  which together with (3.15) yields  $\nabla M(z^*) = 0$ . Otherwise,  $\lim_{k(\in K) \rightarrow \infty} \delta^{j_k} = 0$ . From

step 4 and Theorem 3.1

$$M(z^k + \delta^{j_k} d_k) > C_k - \gamma \|\delta^{j_k} d_k\|^2 \geq M(z^k) - \gamma \|\delta^{j_k} d_k\|^2, \quad (3.17)$$

i.e.,

$$\frac{M(z^k + \delta^{j_k} d_k) - M(z^k)}{\delta^{j_k}} + \gamma \|\delta^{j_k} d_k\|^2 > 0. \quad (3.18)$$

Now that  $M(z)$  is continuously differentiable at  $z^*$ , so we have

$$\nabla M(z^*)^T d^* \geq 0. \quad (3.19)$$

Then, from (3.16), we can get  $\nabla M(z^*)^T d^* = 0$  and

$$\begin{aligned} (d^*)^T \left( H'(z^*)^T H'(z^*) + 2\theta C^* I \right) d^* &= -\nabla M(z^*)^T \left[ H'(z^*)^T H'(z^*) + 2\theta C^* I \right]^{-1} \left( H'(z^*)^T H'(z^*) + 2\theta C^* I \right) d^* \\ &= -\nabla M(z^*)^T d^* = 0. \end{aligned}$$

Since the matrix  $H'(z^*)^T H'(z^*) + 2\theta C^* I$  is a positive matrix, so we have

$$d^* = -\left[ H'(z^*)^T H'(z^*) + 2\theta C^* I \right]^{-1} \nabla M(z^*)^T = 0. \quad (3.20)$$

Now that the matrix  $\left[ H'(z^*)^T H'(z^*) + 2\theta C^* I \right]^{-1}$  is also positive matrix, we can get  $\nabla M(z^*) = 0$ .

#### 4. Numerical experiments

In this section, we carry out some numerical experiments on the LWCP by Algorithm 3.1. All experiments were conducted on a ThinkPad480 with a 1.8GHz CPU and 8.0GB RAM. The codes are run in MATLAB R2018b under Win10.

We first generate the matrices  $P, Q, R$  and vector  $a$  by following way:

$$P = \begin{pmatrix} A \\ M \end{pmatrix}, Q = \begin{pmatrix} 0 \\ I \end{pmatrix}, P = \begin{pmatrix} 0 \\ -A^T \end{pmatrix}, a = \begin{pmatrix} b \\ -f \end{pmatrix}, \quad (4.1)$$

where  $A \in R^{m \times n}$  is a full row rank matrix with  $m < n$ , the matrix  $M$  is an  $n \times n$  symmetric semidefinite matrix,  $b \in R^m$ ,  $f \in R^n$ . In our algorithm we set:  $\gamma = 0.01$ ,  $\sigma = 0.5$ ,  $\delta = 0.8$ ,  $\theta = 10^{-4}$ . The initial points are choosing as:  $x^0 = (1, \dots, 1)$ ,  $s^0 = (1, \dots, 1)$ ,  $y^0 = (0, \dots, 0)$ .

In the course of experiments, we generate LWCP (1.3) by the following two ways.

(i) We take  $A = \text{randn}(m, n)$  with  $\text{rank}(A) = m$ , and  $M = \frac{BB^T}{\|BB^T\|}$  with  $B = \text{rand}(n, n)$ . we first generate

$\hat{x} = \text{rand}(n, 1)$ ,  $f = \text{rand}(n, 1)$ , then we set  $\hat{b} := A\hat{x}$ ,  $\hat{s} = M\hat{x} + f$ ,  $w = \hat{x}\hat{s}$ .

(ii) We choose  $a = \begin{pmatrix} b \\ -f \end{pmatrix} - \xi$  where  $\xi \in R^{n+m}$  is a noise. We choose  $M = \text{diag}(v)$  with  $v = \text{rand}(n, 1)$ .

The matrix  $A$  and vectors  $b, f$  are generated in the same way as (i). In the course of experiments, we take  $\xi = 10^{-4} \text{rand}(1, 1)p$  with  $p := (1, 1, 0, \dots, 0)^T \in R^{n+m}$ .

First, in order to observe the local convergence of algorithm 3.1, we conducted two sets of random test experiments on LWCP (i) with  $n = 1000$ ,  $m = 500$ . Figure 1 gives the convergence curve of  $\|H(z^k)\|$  at the  $k$ -th iteration. We can clearly see that algorithm 3.1 is locally fast, or at least locally superlinear.

Next, we conducted comparative experiments with [13]. In the course of experiments, the parameters in the WCP functions  $\phi_{\tau, q}^w$  are respectively taken as  $\tau = 0.5, q = 3$ ,  $\tau = 1, q = 3$  and  $\tau = 0.3, q = 3$ ,  $\tau = 0.8, q = 3$ . The numerical results are presented in Tables 1, 2, Figures 2 and 3 respectively. Where AIT, ACPU, ANH are respectively the average number of iterations, the average CPU time (unit seconds), and the average number  $\|H(z^k)\|$  of iterations at the end of 10 random experiments. LM represents our experimental result, TLM is the experimental result of [13].

Tables 1 and 2 show the numerical results for LWCP (i). Where, the parameters are taken as  $\tau = 0.5, q = 3$ ;  $\tau = 1, q = 3$  respectively. It can be seen from the table that no matter what value  $\tau$  takes, our algorithm 3.1 has less iteration time or higher accuracy than algorithm 1 in [13].

Figures 2 and 3 show the numerical results for solving LWCP (ii). Where, the parameters are respectively taken as  $\tau = 0.3, q = 3, m = \frac{n}{2}$ ;  $\tau = 0.8, q = 3, m = \frac{n}{2}$ . It can be seen from the figure that with the increase of dimension, the AIT of algorithm 3.1 fluctuates slightly, but it is always smaller than the AIT in [13]. The ACPU increases steadily and always smaller than the ACPU in [13].

When  $\tau = 0.6, q = 3, m = \frac{n}{2}$ , Figure 4 shows the ACPU and AIT comparison line graphs for LWCP (i) and LWCP (ii) solved by algorithms 3.1 and [13] respectively. It can be seen from the figure that after adding noise to LWCP (i), the solution speed of both algorithms decreases, but our algorithm still has certain advantages.

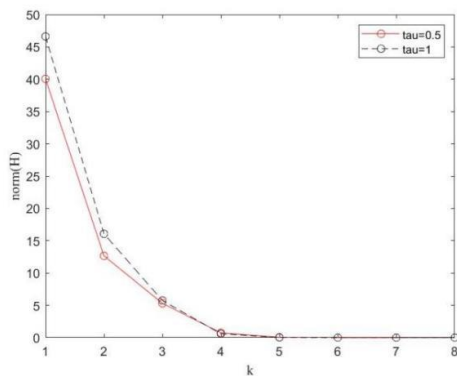
In general, the problems generated by numerical experiments converge in a few iterations. The number of iterations varies slightly with the dimension of the problem. Our algorithm is effective for the linear weighted complementarity problem LWCP (1.3), because each problem can be successfully solved in a very short time with a small number of iterations. Numerical results show the feasibility and effectiveness of the algorithm 3.1.

**Table 1.** Numerical results of solving LWCP (i) ( $\tau = 0.5, q = 3$ ).

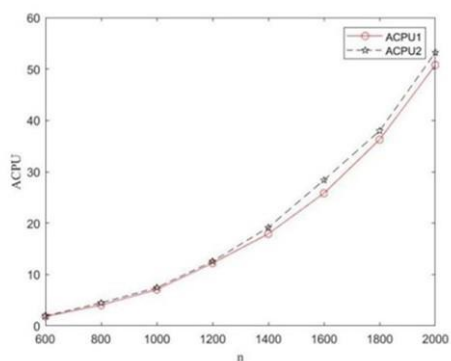
m	n	LM			TLM		
		AIT	ACPU	ANH	AIT	ACPU	ANH
200	500	7.9	0.6960	$5.0131 \times 10^{-12}$	8.0	0.7015	$6.0903 \times 10^{-13}$
		7.8	0.6974	$5.5794 \times 10^{-12}$	7.7	0.6906	$1.2630 \times 10^{-11}$
		7.6	0.6703	$8.5289 \times 10^{-12}$	7.9	0.7025	$3.5098 \times 10^{-13}$
400	800	8.1	2.4705	$5.5548 \times 10^{-13}$	8.8	3.1241	$5.2707 \times 10^{-13}$
		8.2	2.5097	$7.6171 \times 10^{-13}$	8.9	2.6100	$2.2961 \times 10^{-13}$
		8.2	2.6300	$2.4813 \times 10^{-13}$	8.1	2.4039	$3.6750 \times 10^{-12}$
500	1000	8.1	4.4569	$1.2136 \times 10^{-12}$	8.1	4.3590	$2.2894 \times 10^{-12}$
		8.4	4.7697	$3.1192 \times 10^{-13}$	8.4	4.4993	$4.5153 \times 10^{-12}$
		8.2	4.8820	$2.7039 \times 10^{-12}$	8.4	4.4767	$9.3738 \times 10^{-13}$
600	1500	7.9	11.2160	$9.7961 \times 10^{-12}$	8.0	11.6639	$1.0240 \times 10^{-12}$
		8.0	11.4230	$1.0008 \times 10^{-13}$	8.0	11.6522	$9.3154 \times 10^{-13}$
		8.0	11.5575	$1.0238 \times 10^{-12}$	7.9	11.4497	$1.0559 \times 10^{-11}$
1000	1500	9.6	18.4934	$5.6351 \times 10^{-12}$	9.5	18.6699	$1.6880 \times 10^{-11}$
		9.9	19.0396	$5.2759 \times 10^{-12}$	11.1	21.6384	$5.9206 \times 10^{-11}$
		8.4	16.3751	$1.2735 \times 10^{-11}$	10.9	21.3177	$7.6313 \times 10^{-12}$

**Table 2.** Numerical results of solving LWCP (i) ( $\tau = 1, q = 3$ ).

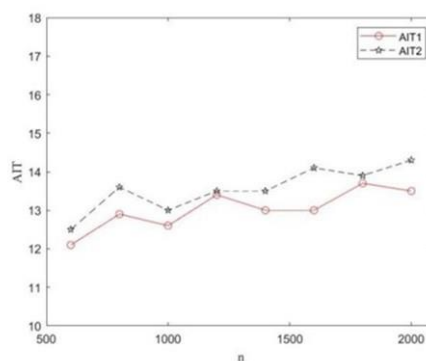
m	n	LM			TLM		
		AIT	ACPU	ANH	AIT	ACPU	ANH
200	500	7.5	0.6642	$1.5973 \times 10^{-11}$	8.0	0.7155	$9.7275 \times 10^{-12}$
		7.4	0.6675	$8.8485 \times 10^{-12}$	8.4	0.7429	$8.7482 \times 10^{-13}$
		7.6	0.6661	$2.5321 \times 10^{-12}$	8.1	0.7167	$6.7422 \times 10^{-13}$
400	800	8.0	2.3642	$2.4919 \times 10^{-13}$	8.8	2.6212	$4.6000 \times 10^{-12}$
		8.0	2.3791	$4.5892 \times 10^{-13}$	8.2	2.5740	$2.4604 \times 10^{-13}$
		8.2	2.4293	$9.2368 \times 10^{-13}$	9.0	2.6885	$1.3216 \times 10^{-12}$
500	1000	8.0	4.3592	$6.2691 \times 10^{-13}$	8.3	4.5328	$3.5736 \times 10^{-12}$
		8.1	4.3174	$3.1221 \times 10^{-13}$	8.2	4.3540	$3.2290 \times 10^{-13}$
		7.9	4.2089	$9.7440 \times 10^{-12}$	9.9	5.3469	$6.5691 \times 10^{-12}$
600	1500	7.9	11.3807	$9.2057 \times 10^{-12}$	8.9	12.9567	$9.7825 \times 10^{-13}$
		7.8	11.2766	$1.3435 \times 10^{-11}$	8.0	11.6116	$9.9437 \times 10^{-13}$
		8.0	11.5494	$9.8875 \times 10^{-13}$	9.2	13.3792	$1.0247 \times 10^{-12}$
1000	1500	9.3	17.6422	$7.8120 \times 10^{-12}$	8.9	17.2609	$3.4824 \times 10^{-12}$
		8.7	16.3247	$4.9019 \times 10^{-12}$	8.8	17.3407	$4.7999 \times 10^{-11}$
		9.3	18.1968	$7.8112 \times 10^{-12}$	9.4	18.4024	$1.3738 \times 10^{-11}$



**Figure 1.** Convergence curve of  $\|H(z^k)\|$  at the  $k$ -th iteration.

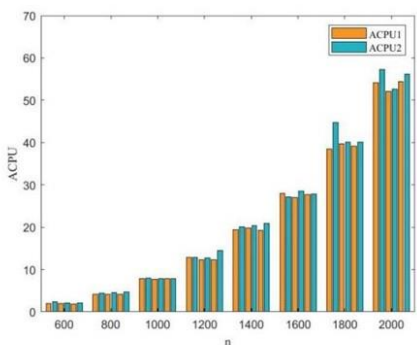


(a) Average CPU time

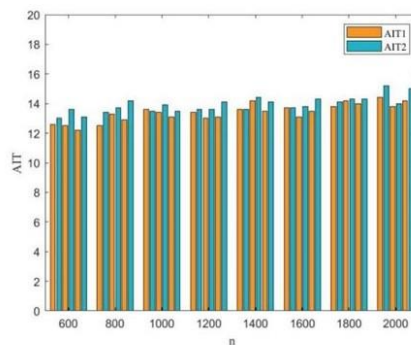


(b) Average number of iterations

**Figure 2.** Comparison curves of solving LWCP (ii) ( $\tau = 0.3, q = 3, m = \frac{n}{2}$ ).

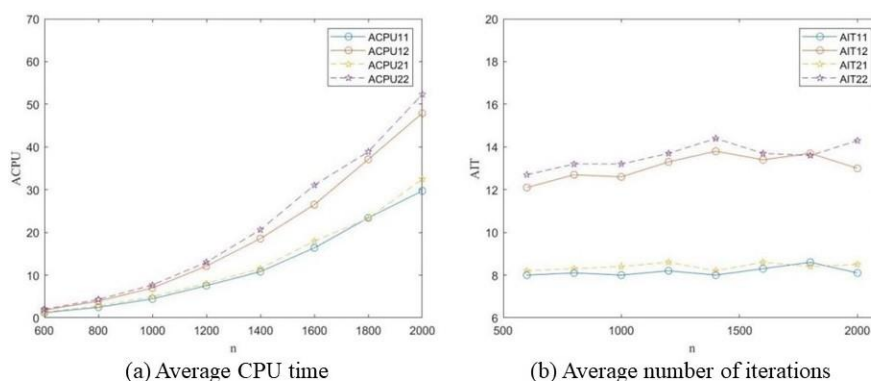


(a) Average CPU time



(b) Average number of iterations

**Figure 3.** Comparison bars of solving LWCP (ii) ( $\tau = 0.8, q = 3, m = \frac{n}{2}$ ).



**Figure 4.** Comparison curves of solving LWCP (i) and LWCP (ii) ( $\tau = 0.6, q = 3, m = \frac{n}{2}$ ).

## 5. Conclusions

Based on the idea of L-M method, with the help of a new class of WCP functions  $\phi_{\tau,q}^c(a,b)$ , we give the algorithm 3.1 for solving the LWCP (1.3). Under certain conditions, our algorithm can obtain the approximate solution of LWCP (1.3). Numerical experiments show the feasibility and effectiveness of the algorithm 3.1.

## Conflict of interest

The authors declare no conflicts of interest.

## References

1. Z. S. Yu, Y. Qin, A cosh-based smoothing Newton method for  $P_0$  nonlinear complementarity problem, *Nonlinear Anal. Real.*, **12** (2011), 875–884. <https://doi.org/10.1016/j.nonrwa.2010.08.012>
2. Z. S. Yu, Z. L. Wang, K. Su, A double nonmonotone Quasi-Newton method for nonlinear complementarity problem based on piecewise NCP functions, *Math. Probl. Eng.*, 2020. <https://doi.org/10.1155/2020/6642725>
3. K. Ueda, N. Yamashita, Global complexity bound analysis of the Levenberg-Marquardt method for nonsmooth equations and its application to the nonlinear complementarity problem, *J. Optim. Theory Appl.*, **152** (2012), 450–467. <https://doi.org/10.1007/s10957-011-9907-2>
4. J. L. Zhang, X. S. Zhang, A smoothing Levenberg - Marquardt method for NCP, *Appl. Math. Comput.*, **178** (2006), 212–228. <https://doi.org/10.1016/j.amc.2005.11.036>
5. J. H. Alcantara, J. S. Chen, A new class of neural networks for NCPs using smooth perturbations of the natural residual function, *J. Comput. Appl. Math.*, **407** (2022), 114092. <https://doi.org/10.1016/j.cam.2022.114092>

6. F. A. Potra, Weighted complementarity problems - a new paradigm for computing equilibria, *SIAM J. Optim.*, **22** (2012), 1634–1654. <https://doi.org/10.1137/110837310>
7. Y. Y. Ye, A path to the arrow-debreu competitive market equilibrium, *Math. Program.*, **111** (2008), 315–348. <https://doi.org/10.1007/s10107-006-0065-5>
8. S. Asadi, Z. Darvay, G. Lesaja, N. Mahdavi-Amiri, F. A. Potra, A full-Newton step interior-point method for monotone weighted linear complementarity problems, *J. Optim. Theory Appl.*, **186** (2020), 864–878. <https://doi.org/10.1007/s10957-020-01728-4>
9. X. N. Chi, G. Q. Wang, A full-Newton step infeasible interior-point method for the special weighted linear complementarity problem, *J. Optim. Theory Appl.*, **190** (2021), 108–129. <https://doi.org/10.1007/s10957-021-01873-4>
10. J. Y. Tang, H. C. Zhang, A nonmonotone smoothing Newton algorithm for weighted complementarity problem, *J. Optim. Theory Appl.*, **189** (2021), 679–715. <https://doi.org/10.1007/s10957-021-01839-6>
11. Z. Y. Liu, J. Y. Tang, A new smoothing-type algorithm for nonlinear weighted complementarity problem, *J. Appl. Math. Comput.*, **64** (2020), 215–226. <https://doi.org/10.1007/s12190-020-01352-5>
12. J. Zhang, A smoothing Newton algorithm for weighted linear complementarity problem, *Optim. Lett.*, **10** (2016), 499–509. <https://doi.org/10.1007/s11590-015-0877-4>
13. J. Y. Tang, J. C. Zhou, Quadratic convergence analysis of a nonmonotone Levenberg-Marquardt type method for the weighted nonlinear complementarity problem. *Comput. Optim. Appl.*, **80** (2021), 213–244. <https://doi.org/10.1007/s10589-021-00300-8>
14. X. H. Liu, W. Wu, Coerciveness of some merit functions over symmetric cones, *J. Ind. Manag. Optim.*, **5** (2009), 603–613. <https://doi.org/10.3934/jimo.2009.5.603>
15. Z. H. Huang, J. Y. Han, D. C. Xu, L. P. Zhang, The non-interior continuation methods for solving the  $P_0$  function nonlinear complementarity problem, *Sci. Chin. Series Math.*, **44** (2001), 1107–1114. <https://doi.org/10.1007/BF02877427>
16. P. Jin, C. Ling, H. F. Shen, A smoothing Levenberg-Marquardt algorithm for semi-infinite programming, *Comput. Optimiz. Appl.*, **60** (2015), 675–695. <https://doi.org/10.1007/s10589-014-9698-0>
17. J. L. Zhang, J. Chen, A smoothing Levenberg-Marquardt type method for LCP, *J. Comput. Math.*, **22** (2004), 735–752.
18. W. A. Liu, C.Y. Wang, A smoothing Levenberg - Marquardt method for generalized semi-infinite programming, *Comput. Appl. Math.*, **32** (2013), 89–105. <https://doi.org/10.1007/s40314-013-0013-y>

